

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-305589
 (43)Date of publication of application : 22.11.1996

(51)Int.Cl. G06F 9/46
 B41J 29/38

(21)Application number : 07-110935
 (22)Date of filing : 09.05.1995

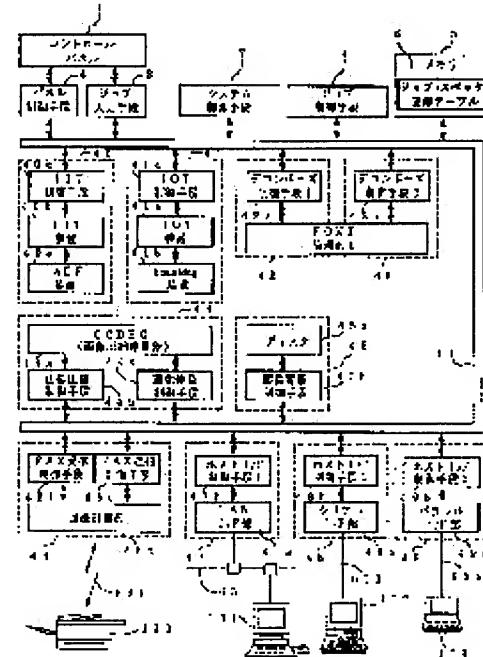
(71)Applicant : FUJI XEROX CO LTD
 (72)Inventor : TOIKAWA YUJI

(54) IMAGE PROCESSING UNIT

(57)Abstract:

PURPOSE: To allocate a resource suitable for a job provided newly with which a new function addition as well as version-up are properly met.

CONSTITUTION: Each of job execution tasks started by a system control means 7 such as an IIT control means 40c, an IOT control means 41c, a FAX reception control means 46b, a FAX transmission control means 46c, a decompose communication 42a, and a decompose control means 43a registers a processed image processing function and a data input output from to a job specification registration table 6. A job control means 4 based on the information selects a task required for the execution of the job based on the content of the job to decide the execution sequence to decide whether or not the job is carried out. Furthermore, a dead lock by simultaneous execution of tasks is prevented by registering an exclusive condition limiting other task to the job specification registration table 6 during the execution of the image processing by each task.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-305589

(43)公開日 平成8年(1996)11月22日

(51)Int.Cl.⁶
G 0 6 F 9/46
B 4 1 J 29/38

識別記号 340

F I
G 0 6 F 9/46
B 4 1 J 29/38

技術表示箇所
340 F
Z

審査請求 未請求 請求項の数 3 O.L (全 17 頁)

(21)出願番号 特願平7-110935

(22)出願日 平成7年(1995)5月9日

(71)出願人 000005496

富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号

(72)発明者 橋川 有史
神奈川県海老名市本郷2274番地 富士ゼロ
ックス株式会社内

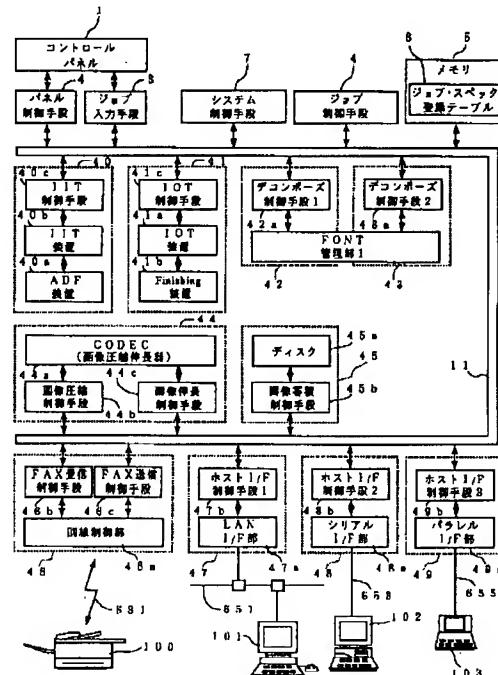
(74)代理人 弁理士 川▲崎▼ 研二 (外1名)

(54)【発明の名称】 画像処理装置

(57)【要約】

【目的】 新たな機能追加や、バージョンアップにも適切に対応させ、新たに提供されるジョブに適切な資源を割当てる。

【構成】 システム制御手段7によって起動された各ジョブ実行系タスク、すなわち I I T 制御手段4 0 c、I O T 制御手段4 1 c、F A X 受信制御手段4 6 b、F A X 送信制御手段4 6 c、デコンポーズ制御手段4 2 a、デコンポーズ制御手段4 3 a等は、処理可能な画像処理機能とデータ入出力形態とをジョブ・スペック登録テーブル6に登録しておく。ジョブ制御手段4は、その情報に基づいて、ジョブ内容から当該ジョブの実行に必要なタスクを選択して実行順序を決定し、ジョブ実行可能であるか否かを判断する。また、各タスクが画像処理を実行中に他のタスクを制限する排他条件を上記ジョブ・スペック登録テーブル6に登録することで、タスク同士の同時実行によるデッドロックを防止する。



【特許請求の範囲】

【請求項1】 複数のタスクの各々で行う画像処理機能および画像データの入出力形態が格納されるジョブ・スペック登録テーブルと、前記画像データに対するジョブをジョブ要求として受け取るジョブ入力手段と、前記ジョブ・スペック登録テーブルに登録されたジョブ・スペックと前記ジョブ入力手段により受け取ったジョブ要求の内容に基づいて、実行させるタスクを選択するタスク選択手段と、前記タスク選択手段により選択されたタスクの実行順序を決定するタスク実行順序決定手段と、前記タスク実行順序決定手段により決定されたタスクによるジョブが実行可能であるか否かを判断するジョブ実行判断手段と、前記ジョブ実行判断手段による判断結果に基づいて、前記タスク実行順序決定手段により決定されたタスクのジョブを実行するタスク実行制御手段とを具備することを特徴とする画像処理装置。

【請求項2】 前記ジョブ・スペック登録テーブルは、前記画像処理機能、およびデータの入出力形態に加えて、前記複数のタスクの各々のジョブ実行中に他のタスクの動作を制限する排他条件を格納し、さらに、現在実行中のジョブのタスクに対する前記排他条件に基づいて、現在実行中のジョブが同時実行可能であるか否かを判断するジョブ同時実行判断手段を備え、前記タスク実行制御手段は、前記ジョブ同時実行判断手段による判断結果に基づいてジョブの同時実行を制御することを特徴とする請求項1記載の画像処理装置。

【請求項3】 前記排他条件を登録する際には、前記複数のタスクを識別するための識別子、または前記複数のタスクをその特徴毎にクラス分けするためのカテゴリ識別子の少なくともいずれか一方を指定し、前記ジョブ実行制御手段は、排他条件としてカテゴリ識別子が指定されている場合には、前記カテゴリ識別子で示されるカテゴリに属する全てのタスクの動作を制限することを特徴とする請求項2記載の画像処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、入力される画像データに特定の画像処理を施す画像処理装置に係り、特に複数の利用者が共有して使用する、複数の機能を有する画像処理装置に関する。

【0002】

【従来の技術】 従来より、1台の画像処理装置でコピー機能、FAX機能、プリンタ機能等を実現できるマルチファンクション機では、さまざまなオプション装置およびアプリケーションを選択できるようになっており、機能の追加／削除が利用者の利用形態に応じて変更できるようになっている。また、このような画像処理装置で

は、性能や、操作性の向上のために、装置内で動作するアプリケーションソフトウェアがバージョンアップされることもある。このため、この画像処理装置内で動作する制御ソフトウェアは、必要に応じて新たに追加されるソフトウェアとの整合性を考慮し、バージョンアップや、機能追加にも対応できる構造が必要となっている。

【0003】 例えば、特開平4-301655号に開示されている技術では、複写機に装着されるさまざまなオプション装置に応じた制御プログラムを、各バージョン毎に格納しておき、装着されたオプション装置のバージョン番号に応じて、制御プログラムを選択できる構成となっている。

【0004】 また、特開平4-261551号に開示されている技術では、複写機の複雑な動作パターンを複数のパートジョブに分けて予めROMに登録しておき、1つのジョブは、そのパートジョブの組み合わせによって実現する構成となっており、新たな機能追加時には、上記パートジョブを変更するだけで対応できる構成となっている。

【0005】 このようなマルチファンクションを実現する画像処理装置の場合には、上述したように、バージョンアップや、機能追加時に対応できる構造が必要になるとともに、複数の利用者が同時に画像処理装置を使用した場合のジョブの同時実行にも対応する必要がある。複数のジョブの同時実行としては、例えば、FAX受信文書をメモリ等の記憶装置に格納中に、コピー処理を行うといった処理が考えられる。このようなジョブの同時実行が当該装置で実行できる可能性は、各ジョブに割り当てる資源（スキャナ、印刷処理部、FAX送受信部、メモリ、ディスク等）が競合しないか、もしくは競合しても共有して使用できるかどうかに關係してくる。したがって、限られた資源でジョブの同時実行を可能にさせるためには、適切な資源割当方法が装置内に実装されている必要がある。

【0006】 上述した問題を解決するために、例えば、特開平5-108381号に開示されている技術では、資源をジョブに割り当てる際に、あるジョブAが獲得している資源を他のジョブBが使用したい場合には、ジョブAに資源割当要求を発行して資源を獲得しにいき、ジョブAが資源を解放したくない場合には、資源割当要求に対して拒絶通知をするといった方法が採られている。

【0007】

【発明が解決しようとする課題】 しかしながら、上述した従来の画像処理装置（特開平4-304655号）では、オプション装置の各バージョンに応じた制御プログラムを、画像処理装置の本体側に工場出荷時点での予測されておかなければならぬので、工場出荷時点での予測されていないような新たなオプション装置の追加には対応できないという問題があった。

【0008】 また、特開平4-261551号に開示さ

れている技術でも同様に、複数のパートジョブを予めROMに登録しておかなければならぬので、新たなパートジョブを提供する場合、ROM交換や制御プログラムの変更等が必要となるといった問題があった。

【0009】さらに、ジョブの競合時の資源割当に関する上述した特開平5-108381号に開示されている技術に関しても、複数のジョブ間での資源の取り合いが生じるので、双方が解放したくない別々の資源を使用している場合などに、ジョブのデッドロックが起こる可能性があるため、複雑なジョブの同時実行が困難であるという問題があった。そして、機能追加や、バージョンアップ等で新たな資源が追加され、その資源を使用するジョブが提供可能になった場合に、従来から提供可能なジョブと新たに追加されたジョブとが競合した場合の資源割当方針が明確化できないため、ジョブの同時実行が不可能になるか、もしくは、例え可能であっても著しく生産性が落ちる可能性があるという問題があった。

【0010】この発明は上述した事情に鑑みてなされたもので、新たな機能追加や、バージョンアップにも適切に対応でき、新たなジョブが提供可能になった場合でも、それらのジョブに適切な資源割当ができ、ジョブ競合時にも効率よくジョブ処理を施すことができる画像処理装置を提供することを目的としている。

【0011】

【課題を解決するための手段】上述した問題点を解決するため、請求項1記載の発明では、複数のタスクの各々で行う画像処理機能および画像データの入出力形態が格納されるジョブ・スペック登録テーブルと、前記画像データに対するジョブをジョブ要求として受け取るジョブ入力手段と、前記ジョブ・スペック登録テーブルに登録されたジョブ・スペックと前記ジョブ入力手段により受け取ったジョブ要求の内容とに基づいて、実行させるタスクを選択するタスク選択手段と、前記タスク選択手段により選択されたタスクの実行順序を決定するタスク実行順序決定手段と、前記タスク実行順序決定手段により決定されたタスクによるジョブが実行可能であるか否かを判断するジョブ実行判断手段と、前記ジョブ実行制御手段による判断結果に基づいて、前記タスク実行順序決定手段により決定されたタスクのジョブを実行するタスク実行制御手段とを具備することを特徴とする。

【0012】また、請求項2記載の発明では、前記ジョブ・スペック登録テーブルは、前記画像処理機能、およびデータの入出力形態に加えて、前記複数のタスクの各々のジョブ実行中に他のタスクの動作を制限する排他条件を格納し、さらに、現在実行中のジョブのタスクに対する前記排他条件に基づいて、現在実行中のジョブが同時実行可能であるか否かを判断するジョブ同時実行判断手段を備え、前記タスク実行制御手段は、前記ジョブ同時実行判断手段による判断結果に基づいてジョブの同時実行を制御することを特徴とする。

【0013】また、請求項3記載の発明では、前記排他条件を登録する際には、前記複数のタスクを識別するための識別子、または前記複数のタスクをその特徴毎にクラス分けするためのカテゴリ識別子の少なくともいずれか一方を指定し、前記ジョブ実行制御手段は、排他条件としてカテゴリ識別子が指定されている場合には、前記カテゴリ識別子で示されるカテゴリに属する全てのタスクの動作を制限することを特徴とする。

【0014】

10 【作用】この発明によれば、各画像処理機能毎に独立したタスクは、それぞれの処理可能な画像処理機能とデータ入出力形態とをジョブ・スペックとしてジョブ・スペック登録テーブルに登録する。タスク選択手段は、ジョブ・スペック登録テーブルに登録されたジョブ・スペックとジョブ入力手段により受け取ったジョブ要求の内容とに基づいて、実行させるタスクを選択する。次いで、タスク実行順序決定手段は、タスク選択手段により選択されたタスクの実行順序を決定し、ジョブ実行判断手段は、タスク実行順序決定手段により決定されたタスクによるジョブが実行可能であるか否かを判断する。そして、タスク実行制御手段は、ジョブ実行制御手段による判断結果に基づいて、タスク実行順序決定手段により決定されたタスクのジョブを実行する。

20 【0015】また、各タスクがそれぞれの画像処理を実行中に、他のタスクの動作を制限する排他条件を登録することで、ジョブが競合した時に、資源の取り合いになる可能性があるタスク同士を相互排他しておく。そして、ジョブ実行制御手段は、要求されたジョブに必要となるタスクが現在実行中のジョブを処理しているタスクの排他条件に合致しているか否かを判断し、合致している場合には、要求されたジョブの実行を待ち、合致していないければジョブを同時実行させる。これにより、新たな機能追加や、バージョンアップにも適切に対応させることが可能となり、新たなジョブが提供可能になった場合でも、それらのジョブに適切な資源を割り当てることが可能となり、ジョブ競合時にも効率よくジョブ処理を施すことが可能となる。

【0016】

【実施例】次に図面を参照してこの発明の一実施例について説明する。

A. 実施例の構成

A-1. ブロック構成

図1は本発明の実施例による画像処理装置の構成を示すブロック図である。図において、コントロールパネル1は、利用者からの操作指示を受け付けたり装置の動作状況を表示する。パネル制御手段2は、上記コントロールパネル1を制御する。また、ジョブ入力手段3は、コントロールパネル1からの利用者の処理開始指示を受け、パラメータ等を参照してシステム内で処理すべき画像処理内容を解析し、それをジョブとして扱いジョブ制

御手段4にジョブ要求を送出する。ジョブ制御手段4は、システム外部から入力される画像データに対しての画像処理ジョブを受け付けて実行可能性判断、実行制御、管理等を行う。メモリ5は、画像処理対象データや、システム内部データを保持／記憶し、また、後述する画像処理モジュール40～49、およびジョブ制御手段4が用いるジョブ・スペック登録テーブル6（後述）に対する領域が予め割り当てられている。システム制御手段7は、上述した各部、および以下で述べる画像処理モジュール40～49を制御する。

【0017】画像処理モジュール40～49は、画像データの入出力を行ったり、画像データに特定の処理を施すものである。画像処理モジュール40～49の各々は、本画像処理装置の利用者の必要に応じて、追加／削除が行われる構成であって、各モジュール単位（図中の破線で囲まれたモジュール単位）でシステム・バス11に着脱可能な構成となっている。画像処理モジュール40は、紙媒体上に記録された画情報をスキャンして、システム内のメモリ5に取り込むイメージ入力端末（Image Input Terminal；以下、IITと呼ぶ）制御モジュールであり、スキャンの対象となる紙媒体をIIT装置40bに供給する自動文書搬送装置（Automatic Document Feeder；以下、ADFと呼ぶ）40a、紙媒体上に記録された画情報をスキャンするIIT装置（スキャナ）40b、およびADF40a、IIT装置40bを制御するとともに、パネル制御手段2、ジョブ入力手段3、およびシステム制御手段7との間でインターフェースをとるIIT制御手段40cから構成されている。

【0018】次に、画像処理モジュール41は、メモリ5に展開されたイメージ画像データを記録用紙に印刷するイメージ出力端末（Image Output Terminal；以下、IOTと呼ぶ）制御モジュールであり、記録用紙や、OHPフィルムといった記録媒体にイメージ画像データを印刷するIOT装置41a、該IOT装置41aによって印刷された記録媒体に対してソーティング、スタッキング、ステープルといった仕分処理を施すフィニッシング装置41b、およびIOT装置41a、フィニッシング装置41bを制御するとともに、パネル制御手段2、ジョブ入力手段3、およびシステム制御手段7との間でインターフェースをとるIOT制御手段41cから構成されている。

【0019】次に、画像処理モジュール42、43は、当該画像処理装置と接続されたホストコンピュータから送られてくるプリント指示データや、テキストデータを解釈してイメージデータに変換し、メモリ5に書き込むデコンポーズ／エミュレーションモジュールであり、P C-P R 2 0 1 H、E S C／P、H P-G L等のエミュレーションモードや、Post Script（ポストスクリプト）等のP D L（Page Description Language；ページ記述言語）に対応したデコンポーズを行うとともに

に、システム制御手段7とのインターフェースをとるデコンポーズ制御手段（以下、デコンポーザと呼ぶ）42a、43a、フォントデータ管理部42bから構成されている。

【0020】次に、画像処理モジュール44は、メモリ5に展開されたイメージ画像データを圧縮／伸長するモジュールであり、画像圧縮伸長器（以下、C O D E Cと呼ぶ）44aと、該C O D E C 44aを制御するとともに、システム制御手段7との間でインターフェースをとる画像圧縮制御手段44b、画像伸長制御手段44cから構成されている。

【0021】次に、画像処理モジュール45は、メモリ5に展開された圧縮／非圧縮イメージ画像データおよびテキスト画像データを、2次記憶媒体に蓄積する画像データ蓄積モジュールであり、ハードディスクや、R A Mディスクといった2次記憶媒体（以下、ディスクと呼ぶ）45a、および該ディスク45aを制御するとともに、システム制御手段7との間でインターフェースをとる画像蓄積制御手段45bから構成されている。

【0022】次に、画像処理モジュール46は、電話回線631を介して画像データを送受信して、システム内のメモリ5に画像データを取り込んだり、メモリ5上の画像データを電話回線631に送出したりするF A X送受信モジュールであり、電話回線631を通じて相手側ファクシミリ装置100とプロトコルのやり取りをする回線制御部46a、相手側ファクシミリ装置100から送出されるF A X画像データを受信し、メモリ5に取り込むとともに、それをジョブとして扱い、ジョブ制御手段4にジョブ要求を行ったり、システム制御手段7との間でインターフェースをとるF A X受信制御手段46b、およびメモリ5に展開されている（圧縮）イメージ画像データを相手側ファクシミリ装置100に送信するとともに、システム制御手段7およびパネル制御手段20との間でインターフェースをとるF A X送信制御手段46cから構成されている。

【0023】次に、画像処理モジュール47、48、49は、ホストコンピュータ101、102、103との間でインターフェースをとり、ホストコンピュータ101、102、103から送出される画像データをメモリ5に取り込むホストインターフェース制御モジュールである。画像処理モジュール47は、E t h e r N e t（イーサネット）、T o k e n R i n g（トークンリング）、L o c a l T a l k（ローカルトーク）といったローカル・エリア・ネットワーク（Local Area Network；以下、L A Nと呼ぶ）651を介してプロトコルを送受信し、ホストコンピュータ101との間でインターフェースをとるL A Nインターフェース部47a、およびL A Nインターフェース部47aを制御してホストコンピュータ101から送出される画像データをメモリ5に取り込むとともに、それをジョブとして扱いジョブ制御

手段4にジョブ要求を送出したり、システム制御手段7およびパネル制御手段2との間でインターフェースをとるホストインターフェース制御手段47bから構成されている。

【0024】画像処理モジュール48は、RS-232C(653)等を介してホストコンピュータ102との間でインターフェースをとるシリアルインターフェース部48a、およびシリアルインターフェース部48aを制御してホストコンピュータ102から送出される画像データをメモリ5に取り込むとともに、それをジョブとして扱いジョブ制御手段4にジョブ要求を送出したり、システム制御手段7およびパネル制御手段2との間でインターフェースをとるホストインターフェース制御手段48bから構成されている。画像処理モジュール49は、セントロニクス(655)等を介してホストコンピュータ103との間でインターフェースをとるパラレルインターフェース部49a、およびパラレルインターフェース部49aを制御してホストコンピュータ103から送出される画像データをメモリ5に取り込むとともに、それをジョブとして扱いジョブ制御手段3にジョブ要求を送出したり、システム制御手段7およびパネル制御手段2とのインターフェースをとるホストインターフェース制御手段49bから構成されている。

【0025】A-2. ソフトウェア構成

上述した基本構成のうち、パネル制御手段2、ジョブ入力手段3、システム制御手段7、ジョブ制御手段4、および各画像処理モジュール内のIIT制御手段40c、IOT制御手段41c、デコンポーズ制御手段42a、デコンポーズ制御手段43a、画像圧縮制御手段44b、画像伸長制御手段44c、画像蓄積制御手段45b、FAX受信制御手段46b、FAX送信制御手段46c、ホストインターフェース制御手段47b、48b、49bは、ソフトウェアで実現されるものとする。なお、本実施例では、システム全体の機能モジュールがリアルタイム・マルチタスク・オペレーティング・システム(以下、RTOSと呼ぶ)上で動作する構成を採っており、上記各ソフトウェアモジュールは、それぞれRTOS上で独自に動作するスレッド構造を持っているものとする。したがって、各ソフトウェアモジュールは、中央演算処理ユニット(図示せず)上で時分割に処理される。以下では、上記各ソフトウェアモジュールを「タスク」と呼び、さらに、各画像処理モジュール内のIIT制御手段40c、IOT制御手段41c、FAX受信制御手段46b、FAX送信制御手段46c、ホストインターフェース制御手段47b、48b、49bを総称して「ジョブ実行系タスク」と呼ぶことにする。

【0026】このような構成で実現される本画像処理装置は、ジョブ入力手段3、FAX受信制御手段46b、ホストインターフェース制御手段47b、48b、49bによって取り込まれたジョブをジョブ制御手段4に送

信することにより、ジョブ要求を送出する。ジョブ制御手段4が受け付けたジョブは、いくつかの画像処理モジュール(ジョブ実行系タスク)を順次実行させていくことにより、利用者の所望の出力を提供する。例えば、コピージョブは、IIT制御モジュール40によってメモリ5に取り込まれたイメージ画像データを、IOT制御モジュール41によって記録用紙に印刷する処理を行う。また、FAX送信ジョブは、IIT制御モジュール40によってメモリ5に取り込まれたイメージ画像データを、FAX送受信モジュール46によって相手側ファクシミリ装置100に送信する処理を行う。また、FAX受信ジョブや、ホストコンピュータから送出されるデータのプリントジョブは、FAX送受信モジュール46や、ホストインターフェースモジュール47~49によって、メモリ5に取り込まれた画像データを、圧縮/伸長モジュール44や、デコンポーズ/エミュレーションモジュール42、43によってイメージ画像データに展開し、それをIOT制御モジュール41によって記録用紙に印刷する処理を行う。

【0027】このように、システムに要求されてくるジョブは、複数の画像処理モジュールが行う画像処理ステップのつながりによって実現される。ここでは、この個々の画像処理ステップを総称してジョブ・ステップと呼ぶことにする。つまり、ジョブは、複数のジョブ・ステップから構成され、各ジョブ・ステップは、画像処理モジュールのいずれかから提供される。

【0028】A-3. ジョブ・スペック登録テーブル
次に、各ジョブ実行系タスクが初期化処理時に登録するジョブ・スペックについて説明する。図2は、上述したジョブ・スペック登録テーブル6の構成を示す模式図である。図において、ジョブ・スペック登録テーブル6は、ジョブ・ステップ領域6aとタスク・スペック領域6bから構成されており、システム内の全タスクからアクセスが可能な共有メモリ領域に配置されている。そして、全てのタスクが本ジョブ・スペック登録テーブル6に非同期でアクセスできるようにするために、バイナリ・セマフォ6cが設けられており、あるタスクのテーブルへのアクセス中に他のタスクからのアクセスを排他する構造を探るようになっていている。

【0029】上記ジョブ・ステップ領域6aは、システム制御手段7によって起動された各ジョブ実行系タスク(IIT制御手段40c、IOT制御手段41c、FAX受信制御手段46b、FAX送信制御手段46c、デコンポーズ制御手段42a、デコンポーズ制御手段43a、ホストインターフェース制御手段47b、48b、49b、画像圧縮制御手段44b、画像伸長制御手段44c、画像蓄積制御手段45b)が提供可能なジョブ・ステップの内容等が格納される領域である。このジョブ・ステップ領域6aには、自タスクのタスクID、提供可能なジョブ・ステップの名称、そして、自タスクがジ

ジョブ・ステップ処理中に同時に動作させたくない他のタスクのタスクID（もしくは、タスク・カテゴリ；後述）が、各ジョブ実行系タスクによって、後述する初期化処理時に、排他条件（mutual exclusive ID；以下、mutex IDと呼ぶ）として登録される。

【0030】なお、図2に示すジョブ・ステップ領域6a内に記載されている「（タスク名）」および「制御手段（図1の符号）」は、説明のために付加したものであり、実際には、ジョブ・スペック登録テーブル6に格納されない項目である。図1に示す各制御手段（ジョブ実行系タスク；IIT制御手段40c、IOT制御手段41c、FAX受信制御手段46b、FAX送信制御手段46c、デコンポーズ制御手段42a、デコンポーズ制御手段43a、ホストインターフェース制御手段47b、48b、49b、画像圧縮制御手段44b、画像伸長制御手段44c、画像蓄積制御手段45b）は、ジョブ・ステップ領域6aの「制御手段（図1の符号）」に対応した欄を登録したことを表している。また、「ジョブ・ステップ名称」も実際の格納時には、図示のような文字列ではなく、各文字列に対応して割り振られたコード番号等で格納されるようになっている。図2では、説明のために理解しやすい形態でジョブ・ステップ領域6aを模式化している。

【0031】また、タスク・スペック領域6bには、システム制御手段7によって起動された各ジョブ実行系タスクの固有の情報が格納されるようになっている。この固有の情報には、タスクID、自タスクが同時処理可能な最大ジョブ数、タスクポート情報等がある。ここで、上記タスクポート情報とは、各ジョブ実行系タスクを図3に示すモデルに当てはめ、各ジョブ実行系タスクに対する入力画像データ、および出力画像データ等の情報を「タスクポート（68）」と呼ばれる入出力ポート毎に記述したものである。したがって、ジョブ・ステップを行う全てのタスクは、1つ以上のタスクポート68を持っており、各タスクポート68には、入力、出力、入出力のいずれかの属性と、入出力される画像データのデータ形式とが規定されている必要がある。したがって、タスク・スペック領域6bには、上記タスクポート情報、すなわち、各ジョブ実行系タスクが使用しているタスクポートの総数、および各タスクポート68の入出力属性、タスクポート68で入出力を行う画像データのデータ種別が格納されるようになっている。

【0032】図3では、例として、Post Script（ポストスクリプト）をデコンポーズする機能を持つタスク55（タスクIDは0x1100）のタスクポート情報をモデル化しているが、このタスク55では、タスクポート68の0番と1番の2つを用意していて、タスクポート68の0番からテキスト（Post Script）データを入力（IN）して、デコンポーズの結果、出力されるイメージデータを、タスクポート68の

1番から出力することを表している。この場合、ジョブ・ステップ領域6aには、図2に示す1a（n+1）という情報が格納され、タスク・スペック領域6bには、図2に示す1b（n+1）という情報がセットされる。他のタスクも同様に、自らジョブ・ステップ情報およびタスク・スペック情報が、それぞれの領域に登録されるようになっている。

【0033】B. 実施例の動作

次に、上述した実施例の動作について説明する。

10 B-1. システム初期化処理

次に、システムの電源投入時の初期化処理について説明する。ここで、図4は本実施例の初期化処理の動作を説明するためのフローチャートである。電源投入されると、まず、システム制御手段7およびジョブ制御手段4、パネル制御手段2、ジョブ入力手段3が起動する。そして、起動されたシステム制御手段7は、図4に示した動作フローに従って初期化処理を行う。

【0034】システム制御手段7は、起動後、ステップS10において、直ちにシステムに実装されているハードウェア・チェックを行う。このとき、工場出荷時に実装されているハードウェアのみならず、オプション追加された画像処理モジュール40～49のチェックも行うとともに、その画像処理モジュール40～49を制御するタスク（制御手段を提供するソフトウェアモジュール）のタスク識別子（以下、タスクIDと呼ぶ）や、タスク・オブジェクトの実体を示すアドレス等も獲得する。次に、ステップS11において、上記タスクID等の情報に基づいて、各ジョブ実行系タスク群（IIT制御手段40c、IOT制御手段41c、FAX受信制御手段46b、FAX送信制御手段46c、デコンポーズ制御手段42a、デコンポーズ制御手段43a、ホストインターフェース制御手段47b、48b、49b、画像圧縮制御手段44b、画像伸長制御手段44c、画像蓄積制御手段45b）を起動する。その後、システム制御手段7は、ステップS12において、起動したタスクからの初期化処理完了メッセージを待ち受ける。

【0035】一方、起動された各ジョブ実行系タスクは、ステップS10において、まず、個々の初期化処理を行う。これは、例えば、各ジョブ実行系タスクが制御するハードウェアモジュール（IIT装置40b、ADF装置40a、IOT装置41a、フィニッシング装置41b、回線制御部46a、LANインターフェース部47a、シリアルインターフェース部48a、パラレルインターフェース部49a、CODEC44a、ディスク45a）の初期パラメータ設定や、診断といった固有の処理であり、自タスクが提供すべきジョブ・ステップが実行可能であるかどうかのチェックを含む。このステップS10で問題がなければ、ステップS102へ進み、提供可能なジョブ・ステップに関する情報をジョブ・スペック（後述）として、ジョブ・スペック登録テーブルを登録する。

ーブル6に登録する。自らは、ステップS103において、非同期に発生するイベントを持つループに入る。なお、ここでは、初期化処理完了通知にRTOSで提供するプロセス（スレッド）間通信機能（以下、IPCメッセージと呼ぶ）を用いて、TASK_AVAILメッセージを送出する。システム制御手段7は、このTASK_AVAILメッセージを受信すると、全てのハードウェアに対して、上述した処理（S10～S12）を繰り返し、全タスクの起動が完了したらシステム初期化処理完了とする。そして、システム初期化処理が完了すると、ジョブ実行が可能になる。

【0036】B-2. ジョブ・スペック登録のフロー
各ジョブ実行系タスクは、前述したような情報をジョブ・スペック登録テーブル6に登録していくわけであるが、この登録時の詳細フロー、すなわち、図5は、図4に示すステップS101の詳細な処理を示すフローチャートである。なお、図5に示すフローチャートは、ライブラリ関数等で、各ジョブ実行系タスクに供給されるものとする。

【0037】図5において、まず、ステップS20において、各ジョブ実行系タスクが登録しようとしているジョブ・スペックの各パラメータの値をチェックする。次に、ステップS21において、パラメータが正当であるか否かを判断する。そして、パラメータの正当性が確認された場合には、ステップS21における判断結果が「YES」となり、ステップS22へ進む。ステップS22では、ジョブ・スペック登録テーブル6をロックするために、バイナリ・セマフォ6cを獲得する。このとき、セマフォ6cの獲得に失敗した場合、すなわち、他のタスクによってセマフォ6cが捕らえられた場合には、このステップS22でセマフォ6cが獲得できるまで待ち受ける。次に、バイナリ・セマフォ6cの獲得によりテーブルがロックできたら、ステップS23、S24において、変数iをインクリメントしながら、ジョブ・ステップ領域6a、およびタスク・スペック領域6bの各領域を先頭からサーチしていき、どのタスクも登録していない未使用エリアを探す。そして、未使用エリアが見つかると、ステップS24における判断結果が「YES」となり、ステップS25へ進む。

【0038】ステップS25では、ジョブ・ステップ領域6a、およびタスク・スペック領域6bのそれぞれの領域に、登録すべきパラメータをセットする。次に、ステップS26へ進み、ジョブ・スペック登録テーブル6をアンロック（バイナリ・セマフォ6cを解放）して正常終了する。一方、ジョブ・ステップ領域6a、およびタスク・スペック領域6bの各領域に未使用エリアがなかった場合には、ステップS23における判断結果は「YES」となり、ステップS27へ進み、その旨を示すエラー番号をセットし、エラー終了する。また、ステップS21において、登録すべきパラメータの値が不正

であった場合には、ステップS21における判断結果が「NO」となり、ステップS28へ進み、その旨を示すエラー番号をセットし、エラー終了する。

【0039】B-3. mutex ID設定時のカテゴリ指定

次に、ジョブ・スペック登録時のmutex IDの指定について説明する。図2に示すジョブ・スペック登録テーブル6内には、ジョブ・スペック領域6aに排他条件mutex IDを設定するエリアがある。これは、各ジョブ実行系タスクがそれぞれのジョブ・ステップを実行する際に、他のジョブによる他のタスクの同時動作を規制するために用いるものである。ここで、他のタスクの同時動作を規制する要件として、以下の項目が考えられる。

【0040】a. 他のタスクとハードウェア資源を共有している際に、その資源を2つ以上のタスクが同時に使用すると誤動作が生じたり、システムが破綻する可能性がある場合である。これは、例えば、回線制御部46aを共有するFAX受信制御手段46bとFAX送信制御手段46c、あるいは1つのCODEC44aを共有する画像圧縮制御手段44bと画像伸長制御手段44c等による同時動作の場合がある。

【0041】b. 他のタスクとのハードウェア資源の共有はないが、バッファメモリや、ディスクといった記憶媒体関連の資源を、同時実行する他のタスクと取り合う可能性があり、記憶媒体の容量によっては、資源の枯渀状態が発生し、ジョブがデッド・ロックを引き起こす可能性がある場合である。これは、複数のデコンポーザ／エミュレーションが同時に動作しているような状況で発生する可能性が高い。例えば、メモリ5の容量がイメージ画像データの2ページ分しかないようなシステムにおいて、ホストコンピュータ101～103のうち、2台のホストコンピュータから別々にプリント要求が来た場合に、異なるデコンポーザ42a、43aがそれぞれのジョブを同時に処理したとする。この条件で、ある時点に双方のデコンポーザ42a、43aが1ページずつメモリ5を用いて両面出力のおもて面のイメージデータをそれぞれレンダリングしてしまった場合、デコンポーザ42a、43a双方の裏面用のページバッファ用メモリがメモリ5から確保できないため、2つのジョブは、デッド・ロック状態となる。

【0042】c. ハードウェアや、その他の仕様上の制約から同時動作を禁止する必要がある場合である。これは、ネットワーク651や、電話回線631上のプロトコルの時間的制約、システム外部への応答の時間的制約などから、ジョブ・ステップ実行時にCPUパワーを相当必要とするタスクなどと同時動作すると、応答できなくなる可能性がある場合等である。

【0043】上述した項目に該当するような状態に陥ることを回避するために、mutex IDに排他条件を設

定する。mutexIDには、自タスクのジョブ・ステップ実行時に同時動作させたくないタスクのタスクIDがセットされる。このとき、複数のタスクに対して動作規制を施したい場合には、特定のタスクを示すタスクIDを指定せずに、そのタスクの属するカテゴリを指定することができる。

【0044】このとき、タスクのカテゴリ分けには、図6に示す分類方法を採用することが必要となる。図6に示す例では、システム内に存在する全てのタスクを5つのカテゴリに分類しており、それぞれのカテゴリに属するタスクのタスクIDに特定のタスクIDバンドを設け、そのタスクIDバンド内でタスクIDを付番させている。例えば、デコンポーザのようなテキスト画像データに何らかの画像処理を施す類のタスクは、DPタスク・グループ1というカテゴリに分類され、このカテゴリに属するタスクのタスクIDには、0x1000～0x1fffの間の値が付番される。したがって、図2に示すタスク5(EMU-201H; PC-PR201Hをエミュレートするタスク)や、タスクn+1(PDL-PS)は、それぞれ0x1000、0x1100のタスクIDが割り振られている。これら2つのタスクには、上述した「2」のようなジョブ実行中のデッドロック状態を回避するために、相互の排他条件として、0x1fffが設定されている。この値は、DPタスク・グループ1というカテゴリに属する全てのタスクの同時実行を規制する意味を持つ。

【0045】同様に、図2におけるタスク3(CODE R:画像圧縮制御手段)とタスク4(DECODE R:画像伸長制御手段)には、上述した「1」のような条件から、タスク3(タスクID=0x2000)のmutexIDにはタスク4(タスクID=0x2100)のタスクIDがセットされ、タスク4のmutexIDにはタスク3のタスクIDがそれぞれ排他条件としてセットされている。

【0046】このような排他条件の設定は、各ジョブ実行系タスク毎に設定可能であり、さらに、図4に示すステップS101において、ジョブ・スペック登録テーブルに登録されるため、各タスクの設計時に予め設計者が設定することが可能となっている。したがって、上述した「1」～「3」のような条件が原因となるようなジョブの同時実行を予め回避することが可能となるとともに、特定のオプションモジュールがシステム内に新規追加されたとしても、ジョブの同時実行時の危険性を回避することが可能となる。すなわち、新規追加されるオプションモジュールのタスク(制御手段)の制御方法がジョブ・スペック登録テーブル6に記入されることになるため、ジョブ制御手段4や、システム制御手段7等に、予めオプションモジュールのタスク全ての制御方法をプログラミングしておく必要がなくなる。これは、ジョブ制御手段4が特定の機種、ハードウェア、オプション群

等に影響することがないことを意味し、各ジョブ実行系タスクの増減による機能追加／削減、ジョブ実行系タスクの入れ替えによるバージョンアップ等に柔軟に対応できるという効果が得られる。なお、mutexIDによるジョブ排他処理の詳細は、ジョブ制御手段の動作説明の項で説明する。

【0047】B-4. ジョブ入力手段の動作

次に、ジョブ入力手段3、FAX受信制御手段46b、ホストインターフェース制御手段47b、48b、49bについて詳細に説明する。これらのタスクは、直接／間接的に利用者が指示した画像処理内容をシステム内に入力／把握し、それをジョブとして取扱い、ジョブ制御手段7にジョブ要求を発行する機能を備えている。図7に示すように、当該画像処理装置内に入力される画像データには、さまざまな入力形態があり、それぞれの入力形態に対応したジョブの検出が必要となる。図7には、本画像処理装置が備えるコントロールパネル1からの指示と、システム制御手段7からの指示とを検出し、それによってジョブ要求をジョブ制御手段4に発行するジョブ入力手段A(3)と、FAX受信制御手段46b内にあって電話回線631からの呼び出しに応じて回線プロトコル内からFAX受信であることを検出し、それによってジョブ要求をジョブ制御手段4に発行するジョブ入力手段B(3b)、そして、LAN651や、シリアル・インターフェース653、パラレル・インターフェース655を経由して入力されてくるホスト・コンピュータ101～103からのプリント要求を検出して、それによってジョブ要求をジョブ制御手段4に発行するジョブ入力手段C(3c)を示している。以下では、これら3種類のジョブ入力手段(3, 3b, 3c)について詳細に説明する。

【0048】B-4a. ジョブ入力手段A(3)

ジョブ入力手段A(3)は、複写機のユーザインターフェースに用いられているようなコントロールパネル1からの利用者の指示をジョブとして扱うものである。コントロールパネル1内には、IITでのスキャンを実行する際のパラメータ(読み取り倍率等)設定部110と、IOTでの印刷時に所望の出力結果を得るために必要なパラメータ(記録用紙サイズ、ソータ/スタッカ/ステープル指示等)設定部111、IITでスキャンしたイメージ画像データをFAX送信したり、ポーリング受信したりする際に必要なパラメータ(相手機の電話番号等)設定部112、そして、これらのパラメータ設定後に実行開始を指示するスタートボタン113からのスタート指示を検出すると、パラメータ設定部110～112の各々から必要なパラメータを参照して、それをジョブ要求パラメータとしてジョブ制御手段4にジョブ要求を送出する。ジョブ要求には、IPCメッセージ送信を用いて、メッセージ識別子JOB_REQとジョブ要求パラメータを送出する。

【0049】また、システム制御手段7では、FAXの通信管理レポートや、各種ハードウェアの診断情報等に必要なシステムデータの格納状態を監視していて、それらのシステムデータが一定量蓄積された段階、もしくは、利用者からのスタートボタンの押下によって、レポート作成用パラメータ設定部23によりレポート作成用パラメータを設定する。ジョブ入力手段A(3)は、これら各種レポート出力要求もジョブとして扱い、その場合は、レポート作成用パラメータ設定部23で設定されたパラメータを参照してジョブ要求を出す。

【0050】B-4 b. ジョブ入力手段B(3 b)
ジョブ入力手段B(3 b)は、回線制御部630上で授受されるFAXプロトコルを監視してFAX受信パラメータを検出するFAX受信制御手段46b内のFAX受信パラメータ検出部24からの指示によって動作する。ジョブ入力手段B(3 b)は、FAX受信パラメータ検出部24からの実行指示を検出すると、FAX受信に必要なパラメータを24から参照し、それをジョブとして扱い、ジョブ制御手段4にジョブ要求を送出する。

【0051】B-4 c. ジョブ入力手段C(3 c)
ジョブ入力手段C(3 c)は、LANあるいはシリアル/パラレルインターフェース(650, 652, 654)で授受されるホストコンピュータとのプロトコルを監視してプリント要求時のパラメータを検出するホストプリントパラメータ検出部25からの指示によって動作する。ジョブ入力手段C(3 c)は、ホストプリントパラメータ検出部25からの実行指示を検出すると、ホストコンピュータから要求されているプリント処理に必要なパラメータをホストプリントパラメータ検出部25から参照し、それをジョブとして扱いジョブ制御手段4にジョブ要求を送出する。

【0052】B-5. ジョブ制御手段の動作
次に、ジョブ制御手段4がジョブ入力手段3、あるいはFAX受信制御手段46b内のジョブ入力手段B(3 b)、あるいはホストインターフェース制御手段47B, 48b, 49b内のジョブ入力手段C(3 c)からのジョブ要求イベントを受信したときの動作を詳細に説明する。

【0053】図8は、ジョブ要求イベント受信時のジョブ制御手段4の動作を示すフローチャートである。IPCメッセージJ_O_B_R_E_Qによるジョブ要求イベントにより、ジョブ実行内容を把握したジョブ制御手段4は、まず、ステップS30において、変数jを全ジョブステップ数とする。次に、ステップS31～S36において、変数iをインクリメントしながら、ジョブ・ステップ登録有無のチェック、ジョブ実行が可能か否か、排他条件が合致するか否か等の処理を実行する。まず、ステップS31では、変数iが上記変数jに達したか否かを判断する。そして、変数iが変数jに達していない場合には、ステップS31における判断結果は「NO」と

なり、ステップS32へ進む。ステップS32では、ジョブ実行に必要な各ジョブ・ステップがジョブ・ステップ登録テーブル6に登録されているどうかをチェックする。すなわち、ジョブ実行に必要となる各ジョブ・ステップが、ジョブスペック登録テーブル6内のジョブ・ステップ領域6aのジョブ・ステップ名称のエリアになるか否かをサーチすることによってチェックする。

【0054】そして、ジョブ実行に必要となる各ジョブ・ステップが、ジョブスペック登録テーブル6内のジョブ・ステップ領域6aのジョブ・ステップ名称のエリアに登録されている場合には、ステップS33における判断結果は「YES」となり、ステップS34へ進む。ステップS34では、そのジョブ・ステップを提供するタスクが現時点で、そのジョブ・ステップを実行できるか否かを判断する。これには、ジョブ・ステップ領域6aのタスクIDのエリアを調べることにより、まず、当該ジョブ・ステップを提供するタスクのタスクIDを調べ、そして、そのタスクIDをキーとしてタスク・スペック領域6bをサーチする。タスク・スペック領域6b内に合致するタスクIDの領域があった場合には、そのタスクの実行ジョブ数(現時点で処理しているジョブの数)、および最大ジョブ実行数(当該タスクが処理することができるジョブの最大数)を調べる。この2つの値が、実行ジョブ数<最大ジョブ実行数であったら、このタスクは、ジョブ・ステップを現時点で実行可能であると判断する。

【0055】そして、ジョブ・ステップを提供しているタスクが現時点で実行可能である場合には、ステップS34における判断結果は「YES」となり、ステップS35へ進む。ステップS35では、さらに、mutexIDの排他条件をチェックする。このチェックには、ジョブ・スペック登録テーブル6内のジョブ・ステップ領域6a内のmutex1Dエリアと、後述するステップS41で値を設定するジョブ制御テーブル9を用いる。ジョブ制御テーブル9は、図9に示す構造を持っており、ジョブID、ジョブ・ステータス、排他条件リスト、当該ジョブを実行する総タスク数、各タスクの情報等が格納されるテーブルである。システムに要求されたジョブには、要求順にジョブ識別子(以下、ジョブIDと呼ぶ)が付番され、ジョブは、ジョブID順にジョブ制御テーブル9に登録される。ジョブ・ステータスには、現時点での各ジョブの実行状態(EXECUTING, PENDING, HALTING等)が格納される。また、排他条件リストには、そのジョブを実行する際に動作する各ジョブ・ステップを実行するタスクによってジョブ・スペック登録テーブル6に登録されたmutex1Dのリストが配列として格納される。

【0056】また、ステップS35では、このジョブ制御テーブル9内の実行中のジョブ(ジョブ・ステータスがEXECUTINGのジョブ)を順にサーチし、各実

行中のジョブの排他条件リストに記載された全m u t e x I Dと、要求されたジョブの当該ジョブ・ステップを提供するタスクのタスク I Dとが合致しているか否か、また、要求されたジョブの当該ジョブ・ステップのm u t e x I Dが、実行中の全ジョブ・ステップを提供するタスクのタスク I Dと合致していないか否か、すなわち、ジョブ制御テーブル9内の実行中ジョブのタスク[0]情報～タスク[n]情報に格納されているタスク I Dと照合することにより調べる。

【0057】次に、ステップS36において、条件が合致したか否かを判断する。このとき、合致しているか否かの判断には、以下の条件式を用いる。

- a. (要求されたジョブステップを提供しているタスク I D AND ジョブ制御テーブル内の排他条件リストにある各m u t e x I D) が、要求されたジョブステップを提供しているタスク I Dと一致しているか否か。
- b. (実行中のジョブを処理している各タスクのタスク I D AND 要求されたジョブステップのm u t e x I D) が、実行中のジョブを処理している各タスクのタスク I Dと一致しているか否か。

上記a, bの双方の条件式のいずれにも合致していない場合には、排他条件は不成立となり、現時点で実行中のジョブと要求されたジョブの当該ジョブ・ステップは同時に実行可能ということになり、ステップS36における判断結果は「NO」となり、ステップS31へ戻る。

【0058】上述したステップS31～S36の各チェックは、要求されたジョブを構成する全ジョブ・ステップに対して行われる。そして、全てが実行可能と判断された場合には、ステップS31における判断結果が「Y E S」となり、ステップS37へ進む。ステップS37では、ジョブ・スペック登録テーブル6のタスク・スペック領域6b内の、タスクポート[0]～[n]情報領域を参照する。ジョブ・ステップを提供する各タスクのタスクポート情報をこの領域から調べ、隣接するタスク間同士のタスクポートを接続していくわけだが、接続の際には、以下に示すタスクポート接続チェックを行う。

a. 前タスクのタスクポートと次タスクのタスクポートを接続する場合には、前タスクのタスクポートの入出力フラグがOUT（もしくはIN/OUT兼用）で、次タスクのタスクポートの入出力フラグがIN（もしくはIN/OUT兼用）のものを選択する。

b. 上記選択を行った結果、接続されたタスクポート同士のデータ種別が一致している必要がある。

【0059】次に、ステップS38では、接続が可能であるか否かを判断する。そして、上記2つの条件を満たすタスクポートが隣接するタスク間に存在した場合には、ステップS38における判断結果が「Y E S」となり、ステップS40へ進む。ステップS40では、そのタスクポートを接続する。次に、接続情報として、ジョブ制御テーブル9内のタスク[0]～[n]情報の各領域

に、各タスクのタスクポートに接続された相手先タスクのタスクポート番号等を登録する。また、ステップS41では、各ジョブ・ステップを提供しているタスクの排他条件m u t e x I Dを、ジョブ・スペック登録テーブル6内のジョブ・ステップ領域6aからジョブ制御テーブル9の排他条件リスト内に転記する。そして、全タスク間がタスクポートで接続できた場合には、ステップS42へ進み、要求されたジョブは、実行可能と判断され、ジョブ実行指示を各タスクに発行する。

- 10 【0060】また、ステップS33, S34, S36, S38の各チェックで「偽」の判定がなされた場合は、各ステップにおける判断結果が「NO」となり、ステップS39へ進む。ステップS39では、要求されたジョブが拒絶、あるいは保留（P e n d）される。例えば、ジョブ制御テーブル9内の、ジョブ I D = 3 で示されるジョブの場合には、タスク I D = 0 x 8100 がジョブ I D = 1 で示されるジョブの排他条件リストと合致しており、同時実行することは不可能であるので、ジョブ・ステータスを保留（P E N D I N G）とする。具体的には、ジョブ3は、スキャンしたイメージ画像を圧縮してF A X送信するジョブであり、ジョブ1は、F A X受信したデータを伸長して印刷出力するジョブであるので、2つのジョブを同時実行させることはF A X送信タスクとF A X受信タスクが公衆回線を共有しているため不可能である。また、ジョブ4は、ホストからのプリントデータを201Hエミュレーションでイメージ画像にして印刷出力するジョブで、ジョブ2はホストからのプリントデータをP o s t S c r i p t デコンポーズして印刷出力するジョブであるので、デコンポーザ同士が排他する条件（0 x 1 f f f）に合致するため不可能である。上述したステップS39もしくはステップS42の処理が終了すると、当該処理を終了する。
- 20 【0061】B-6. 他の実施例
- 30 上述した実施例では、ジョブ・スペック登録テーブル6は、ジョブ・ステップ領域6aとタスク・スペック領域6bの2つから構成されているが、これは、1つのタスクが複数のジョブ・ステップを提供できる場合にも対応できるようにするためにあり、そのようなタスクが存在しない場合には、ジョブ・ステップ領域6aとタスク・スペック領域6bを1つのテーブルにまとめることが可能である。また、図4に示すタスク I Dのカテゴリ分類も、システムの性格や、実装上の仕様に応じて変更することが可能であることは言うまでもない。

- 40 【0062】このように、本実施例では、ジョブ・スペック登録テーブル6に登録された情報に基づいて、ジョブ実行制御を行うので、新たな機能追加や、バージョンアップにも適切に対応するジョブ制御手段が実現でき、新たなジョブが提供可能になった場合でも、それらのジョブに適切な資源割当を行うことができ、ジョブ競合時にも効率よくジョブ処理を施すことができる。

【0063】

【発明の効果】以上、説明したように、この発明によれば、ジョブ・スペック登録テーブルに登録された情報に基づいて、ジョブ実行制御を行うので、新たな機能追加や、バージョンアップにも適切に対応するジョブ制御手段が実現でき、新たなジョブが提供可能になった場合でも、それらのジョブに適切な資源割当を行うことができ、ジョブ競合時にも効率よくジョブ処理を施すことができるという利点が得られる。

【図面の簡単な説明】

【図1】 本発明の実施例による画像処理装置の構成を示すブロック図である。

【図2】 本実施例のジョブ・スペック登録テーブルを説明するための模式図である。

【図3】 本実施例のタスクポートを説明するための模式図である。

【図4】 本実施例のシステム初期化処理時の動作を説明するためのフローチャートである。

【図5】 本実施例のジョブ・スペック登録の動作を説明するためのフローチャートである。

【図6】 本実施例の各タスクのカテゴリ分類一覧の例を示す模式図である。

【図7】 本実施例のジョブ入力手段の模式図である。

【図8】 本実施例のジョブ制御手段のジョブ要求時の動作を説明するためのフローチャートである。

【図9】 ジョブ制御テーブルの模式図である。

【符号の説明】

1 コントロールパネル

2 パネル制御手段

3 ジョブ入力手段

4 ジョブ制御手段 (タスク選択手段、タスク実行順序*)

* 決定手段、ジョブ実行判断手段、タスク実行制御手段、ジョブ同時実行判断手段)

5 メモリ

6 ジョブ・スペック登録テーブル

6 a ジョブ・ステップ領域

6 b タスク・スペック領域

7 システム制御手段

40~49 画像処理モジュール (複数のタスク)

40 a 自動文書搬送装置

10 40 b I I T 装置

40 c I I T 制御手段

41 a I O T 装置

41 b フィニッシング装置

41 c I O T 制御手段

42 a デコンポーズ制御手段

42 b フォントデータ管理部

43 a デコンポーズ制御手段

44 a 画像圧縮伸長器

44 b 画像圧縮制御手段

20 44 c 画像伸長制御手段

45 a 2 次記憶媒体

45 b 画像蓄積制御手段

46 a 回線制御部

46 b F A X 受信制御手段

46 c F A X 送信制御手段

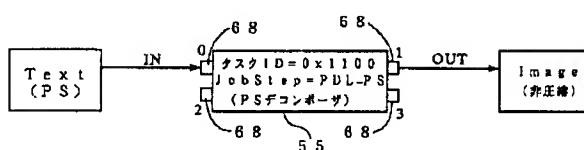
47 a, 48 a, 49 a ホストインターフェース制御手段

47 b L A N インターフェース部

48 b シリアルインターフェース部

30 49 b パラレルインターフェース部

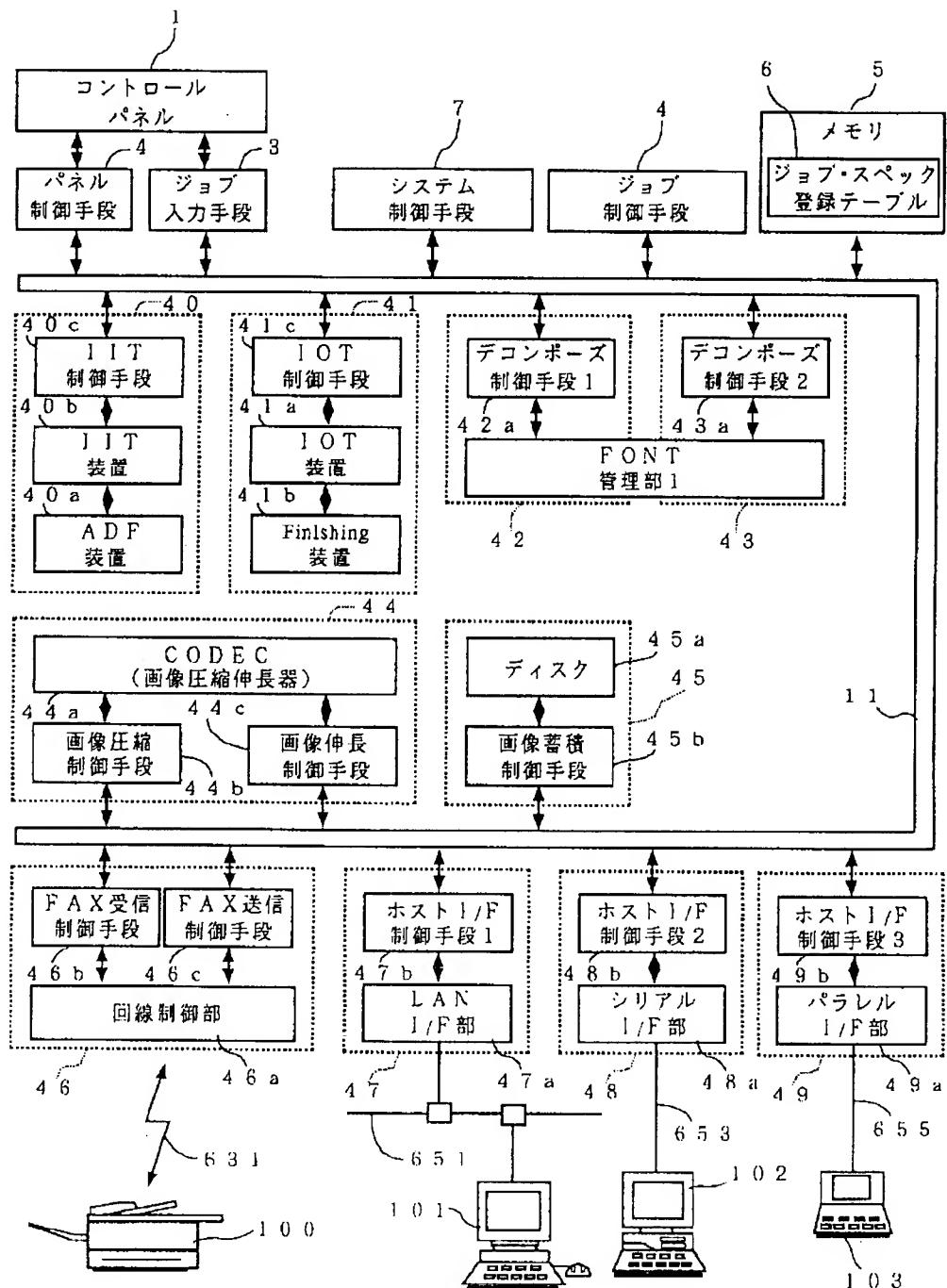
【図3】



【図6】

| カテゴリ | 特徴 | タスク ID バンド |
|--------------|---------------------------------------|---------------|
| システムタスク・グループ | システム制御をする (システム制御手段、ジョブ制御手段、ジョブ入力手段等) | 0x0000~0x0fff |
| IPタスク・グループ1 | テキスト画像データに何らかの画像処理を施す | 0x1000~0x1fff |
| IPタスク・グループ2 | イメージ画像データに何らかの画像処理を施す | 0x2000~0x2fff |
| INタスク・グループ | システム内に画像データを入力する | 0x4000~0x4fff |
| OUTタスク・グループ | システム内の画像データを外部に出力する | 0x8000~0x8fff |

【図1】



【図2】

Diagram illustrating the relationship between Task Control Tables (6a, 6b) and Task Port Tables (1a, 1b).

Table 6a: Task Control Table

| タスク名 | 制御手段 図1の符号 | タスクID | ジョブ ステップ 名称 | mutex ID (排他条件) |
|--------|---------------|--------|-------------------|--------------------|
| タスク1 | 40c | 0x4000 | SCAN-1 | 0x4000 |
| タスク2 | 41c | 0x8000 | PRINT-1 | 0 |
| タスク3 | 44b | 0x2000 | CODER | 0x2100 |
| タスク4 | 46c | 0x2100 | DECODER | 0x2000 |
| タスク5 | 42a | 0x1000 | EMU-201H | 0x1fff |
| タスク6 | 47b | 0x4100 | AppleTalk | 0 |
| ... | | | | |
| タスクn-1 | 46b | 0x4200 | FAX (Recv) | 0x8100 |
| タスクn | 46c | 0x8100 | FAX (Send) | 0x4200 |
| タスクn+1 | 43a | 0x1100 | PDL-PS | 0x1fff |

Table 6b: Task Control Table (Continuation)

| タスク ID | 実行 ジョブ 数 | 最大 ジョブ 数 | タスク ポート 総数 | タスクポート[0] 情報 | タスクポート[1] 情報 | タスクポート[n] 情報 | |
|-----------|----------------|----------------|------------------|-----------------|------------------|-----------------|----------------|
| | | | | 入出力 属性 | データ 種別 | 入出力 属性 | データ 種別 |
| 0x4000 | 0 | 1 | 1 | OUT | Image (非圧縮) | — | — |
| 0x8000 | 0 | 3 | 1 | IN | Image (非圧縮) | — | — |
| 0x2000 | 0 | 1 | 2 | IN | Image (非圧縮) | OUT | Image (圧縮) |
| 0x2100 | 0 | 1 | 2 | IN | Image (圧縮) | OUT | Image (非圧縮) |
| 0x1000 | 0 | 1 | 2 | IN | Text (PR201H) | OUT | Image (非圧縮) |
| 0x4100 | 0 | 5 | 1 | OUT | Text (PS) | — | — |
| ... | | | | | | | — |
| 0x4200 | 0 | 1 | 1 | OUT | Image (圧縮) | — | — |
| 0x8100 | 0 | 1 | 1 | IN | Image (圧縮) | — | — |
| 0x1100 | 0 | 1 | 2 | IN | Text (PS) | OUT | Image (非圧縮) |

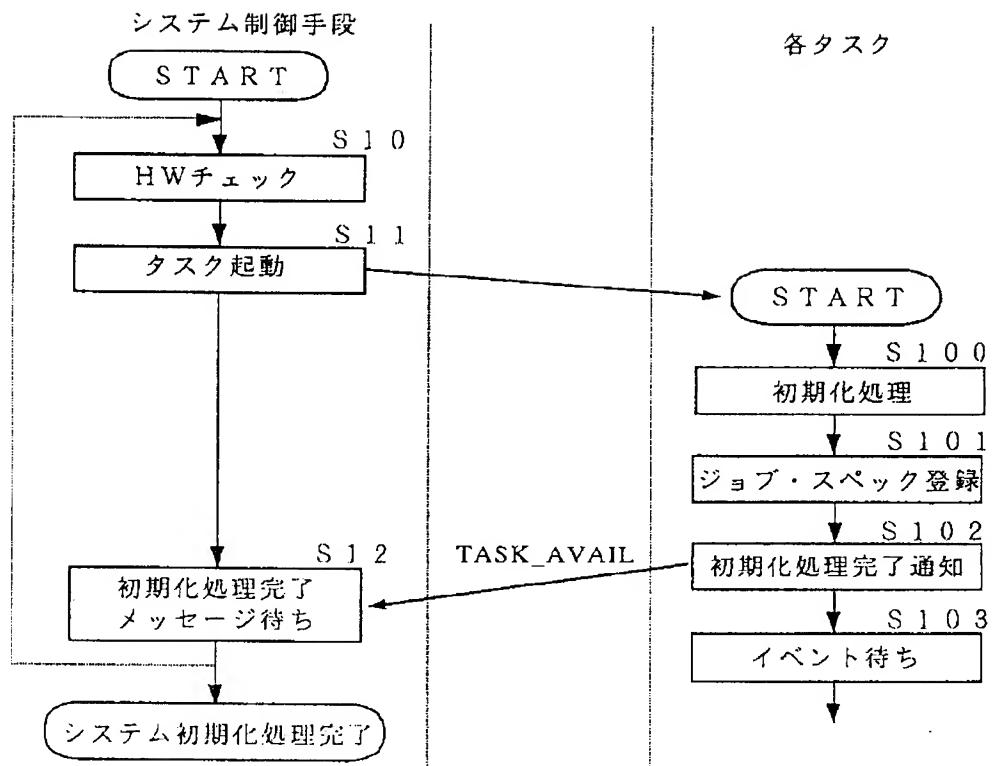
Table 1a: Task Port Table (Continuation)

| タスク ID | 実行 ジョブ 数 | 最大 ジョブ 数 | タスク ポート 総数 | タスクポート[0] 情報 | タスクポート[1] 情報 | タスクポート[n] 情報 | |
|-----------|----------------|----------------|------------------|-----------------|------------------|-----------------|----------------|
| | | | | 入出力 属性 | データ 種別 | 入出力 属性 | データ 種別 |
| 0x4000 | 0 | 1 | 1 | OUT | Image (非圧縮) | — | — |
| 0x8000 | 0 | 3 | 1 | IN | Image (非圧縮) | — | — |
| 0x2000 | 0 | 1 | 2 | IN | Image (非圧縮) | OUT | Image (圧縮) |
| 0x2100 | 0 | 1 | 2 | IN | Image (圧縮) | OUT | Image (非圧縮) |
| 0x1000 | 0 | 1 | 2 | IN | Text (PR201H) | OUT | Image (非圧縮) |
| 0x4100 | 0 | 5 | 1 | OUT | Text (PS) | — | — |

Table 1b: Task Port Table (Continuation)

| タスク ID | 実行 ジョブ 数 | 最大 ジョブ 数 | タスク ポート 総数 | タスクポート[0] 情報 | タスクポート[1] 情報 | タスクポート[n] 情報 | |
|-----------|----------------|----------------|------------------|-----------------|-----------------|-----------------|----------------|
| | | | | 入出力 属性 | データ 種別 | 入出力 属性 | データ 種別 |
| 0x4200 | 0 | 1 | 1 | OUT | Image (圧縮) | — | — |
| 0x8100 | 0 | 1 | 1 | IN | Image (圧縮) | — | — |
| 0x1100 | 0 | 1 | 2 | IN | Text (PS) | OUT | Image (非圧縮) |

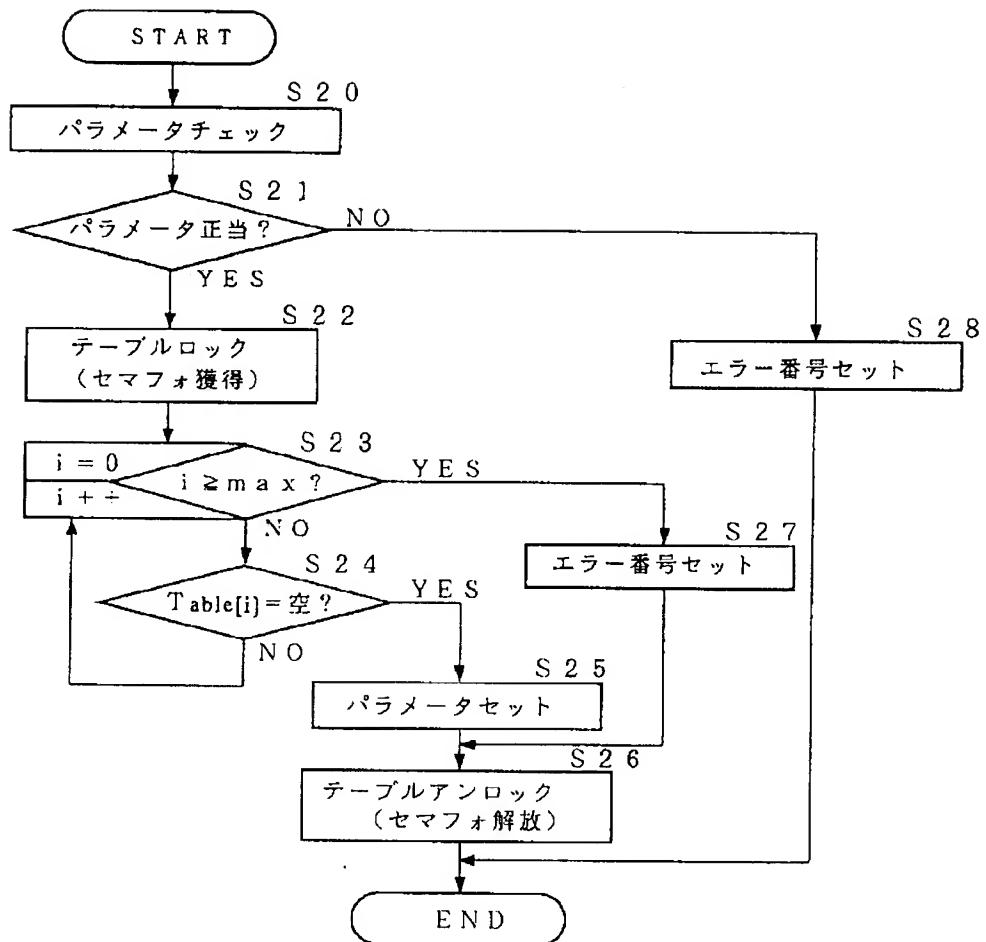
【図4】



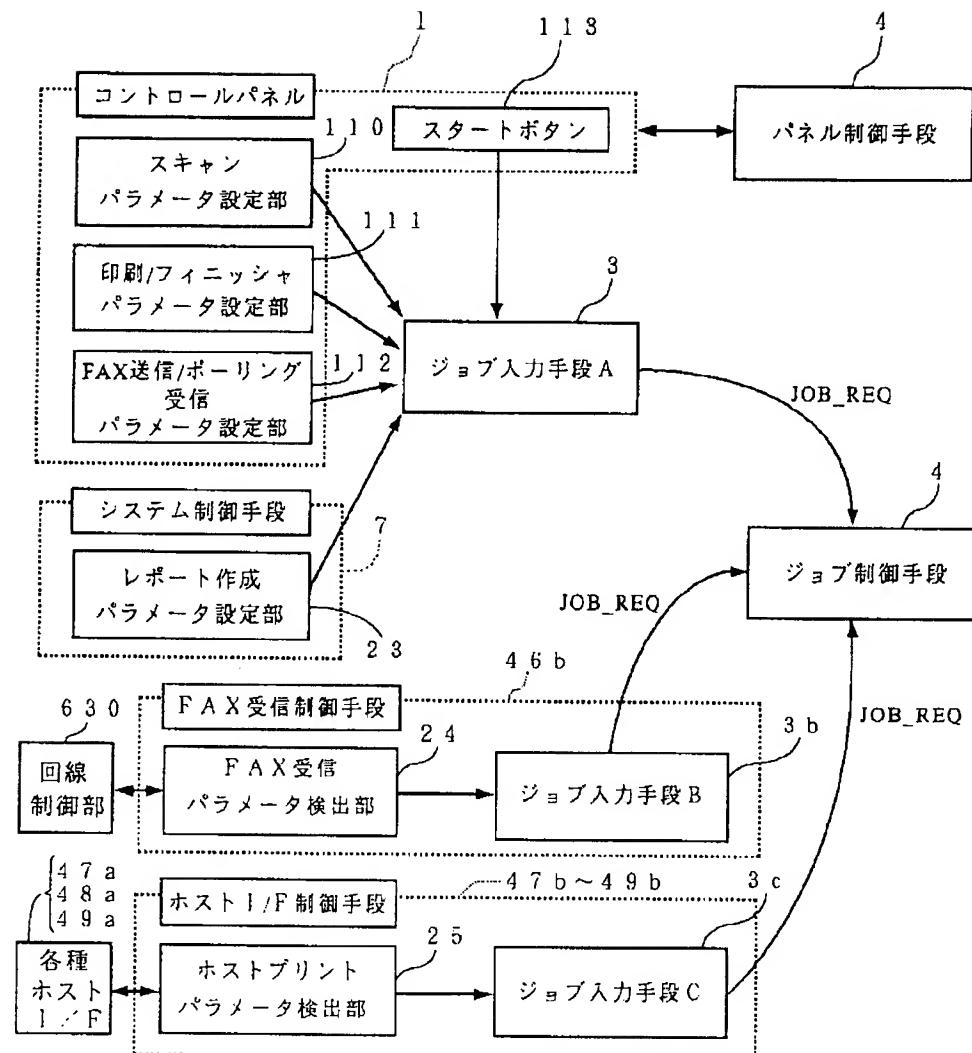
【図9】

| ジョブID | ステータス | 併他条件 (mutex ID) | 総タスク数 | タスク[0] 情報 | タスク[1] 情報 | タスク[2] 情報 | ... | タスク[n] 情報 |
|-------|-----------|-----------------------|-------|---------------|---------------|---------------|-----|--------------|
| 1 | EXECUTING | 0x8100 0x2000 0 | 3 | 0x4200 ... | 0x2100 ... | 0x8000 ... | ... | — |
| 2 | EXECUTING | 0 0x2000 0 | 3 | 0x4100 ... | 0x1100 ... | 0x8000 ... | ... | — |
| 3 | PENDING | | 3 | 0x4000 ... | 0x2000 ... | 0x8100 ... | ... | — |
| 4 | PENDING | | 2 | 0x4300 ... | 0x1000 ... | 0x8000 ... | ... | — |

【図5】



【図7】



【図8】

